Naive Bayes

The Probability of Being Right

Ahmed BADI

Mathematics & Machine Learning Enthusiast

ahmedbadi905@gmail.com
linkedin.com/in/badi-ahmed

December 15, 2025

Overview

Introduction

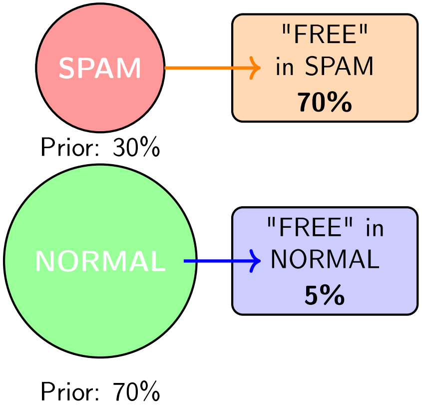
The Intuition: Email Classification

Scenario: You receive an email with the word "FREE"

What we know:

- 70% of SPAM emails contain "FREE"
- 5% of NORMAL emails contain "FREE"
- 30% of emails are SPAM (prior)
- 70% of emails are NORMAL (prior)

Question: Is this email SPAM or NORMAL?



Circle size = Prior probability

Strong evidence for SPAM!

Key Insight

Combine how common SPAM is (Prior) with how typical "FREE" appears in SPAM (Likelihood)!

Bayes' Theorem

The Mathematical Foundation

Thomas Bayes (1700s) gave us the formula to reverse conditional probability.

$$P(y|X) = \frac{P(X|y) \cdot P(y)}{P(X)}$$

- $P(y|X)$ **Posterior:** Probability of Class given Evidence. (Goal)
- $P(X|y)$ **Likelihood:** Probability of Evidence given Class.
- $P(y)$ **Prior:** How common is the class generally?
- $P(X)$ **Evidence:** Normalizing constant.

Visualizing the Theorem

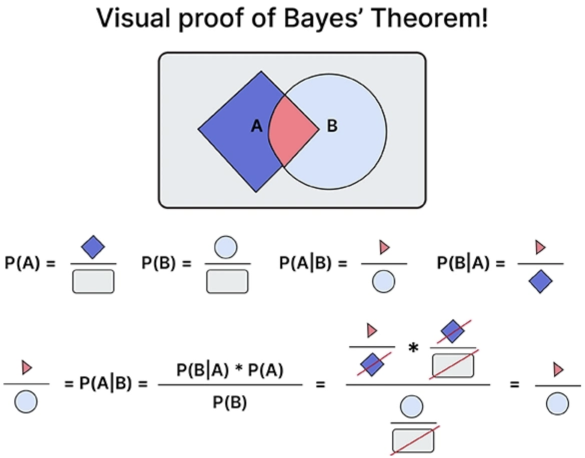


Figure: Updating Prior Beliefs with New Evidence.

The "Naive" Assumption

Why is it called Naive?

The Problem

Calculating $P(x_1, x_2, x_3|y)$ requires knowing how features interact.
Does the word "Money" appear more often with "Free"?

The Assumption

We assume all features are **INDEPENDENT**.

$$P(x_1, \dots, x_n|y) \approx P(x_1|y) \cdot P(x_2|y) \cdot \dots \cdot P(x_n|y)$$

It is "dumb" but computationally magical. It turns an impossible problem into simple multiplication.

Independence Visualization

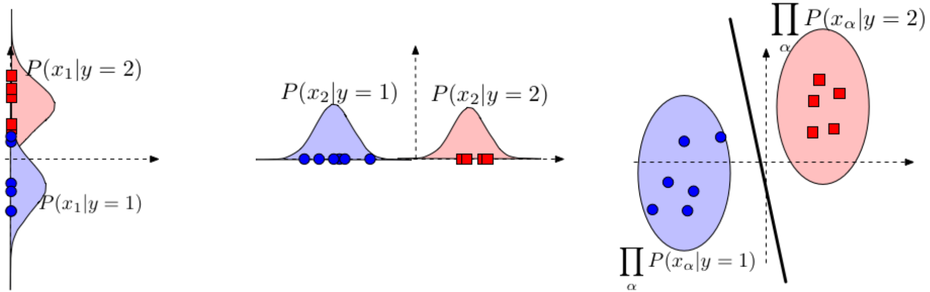


Figure: Features depend on the Class, but NOT on each other.

Variations Challenges

Three Flavors of Naive Bayes

1. **Gaussian**
   - For continuous data (Height, Weight).
   - Assumes Bell Curve.
2. **Multinomial**
   - For counts (Word frequency).
   - Standard for NLP/Spam.
3. **Bernoulli**
   - For binary features (Yes/No).
   - Short texts.

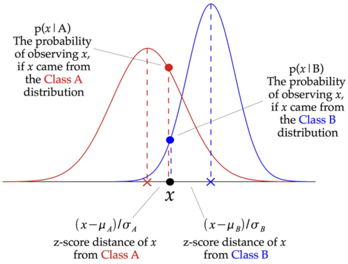


Figure: Gaussian Decision Boundary.

The Zero Frequency Problem

Scenario: A new email has the word "Bitcoin".

- "Bitcoin" was never in the Spam training set.
- $P(\text{"Bitcoin"}|\text{Spam}) = 0$.

$$P(\text{Spam}) = P(\text{"Free"}|S) \times 0 \times \cdots = 0$$

The whole model crashes!

Solution: Laplace Smoothing

Add 1 to every count. Pretend we've seen every word once.

$$P(w|y) = \frac{\text{Count} + 1}{\text{Total} + \text{VocabSize}}$$

Worked Example: Spam Filter

Spam or Ham?

New Message: "Money Mom"

Class: Spam (Prior 0.67)

- P("Money"|S) = High (0.33)
- P("Mom"|S) = Low (0.11)
- **Score:** $0.67 \times 0.33 \times 0.11 = 0.024$

Class: Ham (Prior 0.33)

- P("Money"|H) = Low (0.14)
- P("Mom"|H) = High (0.28)
- **Score:** $0.33 \times 0.14 \times 0.28 = 0.013$

Result: $0.024 > 0.013 \Rightarrow$ SPAM

The Process Flow

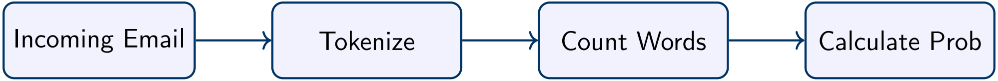


Figure: From Raw Text to Decision.

Conclusion

Summary

Pros

- Blazingly fast.
- Interpretable (Probabilities).
- Great for Text/NLP.
- Works with small data.

Cons

- Independence assumption is false.
- "Zero Frequency" issues.
- Probabilities can be overconfident.

Thank You!

"Sometimes, being naive is the smartest strategy."

Questions?